Assignment-Courier Management System

– Elakkiya

Task1 Database Design

Design a SQL schema for a Courier Management System with tables for Customers, Couriers, Orders, and Parcels. Define the relationships between these tables using appropriate foreign keys.

Requirements:

• Define the Database Schema • Create SQL tables for entities such **as User, Courier, Employee, Location,Payment**

• Define relationships between these tables (one-to-many, many-to-many, etc.).

• **Populate Sample Data** • Insert sample data into the tables to simulate real-world scenarios.

```
mysql> CREATE DATABASE COURIER_MANAGEMENT;
Query OK, 1 row affected (0.01 sec)

mysql> USE COURIER_MANAGEMENT;
Database changed
```

```
mysql> CREATE TABLE COURIER ( COURIERID INT PRIMARY KEY,
    ->     SENDERNAME VARCHAR(255),
    ->     SENDERADDRESS TEXT,
    ->     RECEIVERNAME VARCHAR(255),
    ->     RECEIVERADDRESS TEXT,
    ->     WEIGHT DECIMAL(5,2),
    ->     STATUS VARCHAR(50),
    ->     TRACKINGNUMBER VARCHAR(20) UNIQUE,
    ->     DELIVERYDATE DATE
    -> );
Query OK, 0 rows affected (0.10 sec)

mysql> DESC COURIER;
+-----------------+--------------+------+-----+---------+-------+
| Field           | Type         | Null | Key | Default | Extra |
+-----------------+--------------+------+-----+---------+-------+
| COURIERID       | int          | NO   | PRI | NULL    |       |
| SENDERNAME      | varchar(255) | YES  |     | NULL    |       |
| SENDERADDRESS   | text         | YES  |     | NULL    |       |
| RECEIVERNAME    | varchar(255) | YES  |     | NULL    |       |
| RECEIVERADDRESS | text         | YES  |     | NULL    |       |
| WEIGHT          | decimal(5,2) | YES  |     | NULL    |       |
| STATUS          | varchar(50)  | YES  |     | NULL    |       |
| TRACKINGNUMBER  | varchar(20)  | YES  | UNI | NULL    |       |
| DELIVERYDATE    | date         | YES  |     | NULL    |       |
+-----------------+--------------+------+-----+---------+-------+
9 rows in set (0.01 sec)
```

```
mysql> CREATE TABLE USER;
ERROR 4028 (HY000): A table must have at least one visible column.
mysql> CREATE TABLE USER (
    ->     USERID INT PRIMARY KEY,
    ->     NAME VARCHAR(255),
    ->     EMAIL VARCHAR(255) UNIQUE,
    ->     PASSWORD VARCHAR(255),
    ->     CONTACTNUMBER VARCHAR(20),
    ->     ADDRESS TEXT
    -> );
Query OK, 0 rows affected (0.11 sec)

mysql> DESC USER;
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| USERID        | int          | NO   | PRI | NULL    |       |
| NAME          | varchar(255) | YES  |     | NULL    |       |
| EMAIL         | varchar(255) | YES  | UNI | NULL    |       |
| PASSWORD      | varchar(255) | YES  |     | NULL    |       |
| CONTACTNUMBER | varchar(20)  | YES  |     | NULL    |       |
| ADDRESS       | text         | YES  |     | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
6 rows in set (0.02 sec)
```

```
mysql> CREATE TABLE COURIER_SERVICES (
    ->     SERVICEID INT PRIMARY KEY,
    ->     SERVICENAME VARCHAR(100),
    ->     COST DECIMAL(8,2)
    -> );
Query OK, 0 rows affected (0.09 sec)

mysql> DESC COURIER_SERVICES;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| SERVICEID   | int          | NO   | PRI | NULL    |       |
| SERVICENAME | varchar(100) | YES  |     | NULL    |       |
| COST        | decimal(8,2) | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

```
mysql> CREATE TABLE EMPLOYEE (
    ->     EMPLOYEEID INT PRIMARY KEY,
    ->     NAME VARCHAR(255),
    ->     EMAIL VARCHAR(255) UNIQUE,
    ->     CONTACTNUMBER VARCHAR(20),
    ->     ROLE VARCHAR(50),
    ->     SALARY DECIMAL(10,2)
    -> );
Query OK, 0 rows affected (0.10 sec)

mysql> DESC EMPLOYEE;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| EMPLOYEEID    | int           | NO   | PRI | NULL    |       |
| NAME          | varchar(255)  | YES  |     | NULL    |       |
| EMAIL         | varchar(255)  | YES  | UNI | NULL    |       |
| CONTACTNUMBER | varchar(20)   | YES  |     | NULL    |       |
| ROLE          | varchar(50)   | YES  |     | NULL    |       |
| SALARY        | decimal(10,2) | YES  |     | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
6 rows in set (0.01 sec)
```

```
mysql> CREATE TABLE LOCATION (
    ->     LOCATIONID INT PRIMARY KEY,
    ->     LOCATIONNAME VARCHAR(100),
    ->     ADDRESS TEXT
    -> );
Query OK, 0 rows affected (0.07 sec)

mysql> DESC LOCATION;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| LOCATIONID   | int          | NO   | PRI | NULL    |       |
| LOCATIONNAME | varchar(100) | YES  |     | NULL    |       |
| ADDRESS      | text         | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

```
mysql> CREATE TABLE PAYMENT (
    ->     PAYMENTID INT PRIMARY KEY,
    ->     COURIERID INT,
    ->     LOCATIONID INT,
    ->     AMOUNT DECIMAL(10,2),
    ->     PAYMENTDATE DATE,
    ->     FOREIGN KEY (COURIERID) REFERENCES COURIER(COURIERID),
    ->     FOREIGN KEY (LOCATIONID) REFERENCES LOCATION(LOCATIONID)
    -> );
Query OK, 0 rows affected (0.11 sec)

mysql> DESC PAYMENT;
+-------------+---------------+------+-----+---------+-------+
| Field       | Type          | Null | Key | Default | Extra |
+-------------+---------------+------+-----+---------+-------+
| PAYMENTID   | int           | NO   | PRI | NULL    |       |
| COURIERID   | int           | YES  | MUL | NULL    |       |
| LOCATIONID  | int           | YES  | MUL | NULL    |       |
| AMOUNT      | decimal(10,2) | YES  |     | NULL    |       |
| PAYMENTDATE | date          | YES  |     | NULL    |       |
+-------------+---------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

INSERTING SAMPLE DATA:

```
mysql> INSERT INTO USER VALUES
    -> (1, 'ELAKKIYA', 'ELAKKIYA@GMAIL.COM', 'PASS123', '9000000001', 'CHENNAI'),
    -> (2, 'LAVANYA', 'LAVANYA@GMAIL.COM', 'PASS234', '9000000002', 'MADURAI'),
    -> (3, 'KASHIFA', 'KASHIFA@GMAIL.COM', 'PASS345', '9000000003', 'COIMBATORE'),
    -> (4, 'SHOBITHA', 'SHOBITHA@GMAIL.COM', 'PASS456', '9000000004', 'BANGALORE'),
    -> (5, 'RITHIKA', 'RITHIKA@GMAIL.COM', 'PASS567', '9000000005', 'HYDERABAD');
Query OK, 5 rows affected (0.03 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM USER;
+--------+----------+--------------------+----------+---------------+------------+
| USERID | NAME     | EMAIL              | PASSWORD | CONTACTNUMBER | ADDRESS    |
+--------+----------+--------------------+----------+---------------+------------+
|      1 | ELAKKIYA | ELAKKIYA@GMAIL.COM | PASS123  | 9000000001    | CHENNAI    |
|      2 | LAVANYA  | LAVANYA@GMAIL.COM  | PASS234  | 9000000002    | MADURAI    |
|      3 | KASHIFA  | KASHIFA@GMAIL.COM  | PASS345  | 9000000003    | COIMBATORE |
|      4 | SHOBITHA | SHOBITHA@GMAIL.COM | PASS456  | 9000000004    | BANGALORE  |
|      5 | RITHIKA  | RITHIKA@GMAIL.COM  | PASS567  | 9000000005    | HYDERABAD  |
+--------+----------+--------------------+----------+---------------+------------+
5 rows in set (0.00 sec)
```

```
mysql> INSERT INTO COURIER VALUES
    -> (101, 'ELAKKIYA', 'CHENNAI', 'ROJA', 'DELHI', 3.25, 'IN TRANSIT', 'TRK100001', '2025-06-18'),
    -> (102, 'LAVANYA', 'MADURAI', 'SEREESHA', 'MUMBAI', 2.10, 'DELIVERED', 'TRK100002', '2025-06-08'),
    -> (103, 'KASHIFA', 'COIMBATORE', 'HARITHA', 'PUNE', 1.80, 'IN TRANSIT', 'TRK100003', '2025-06-20'),
    -> (104, 'SHOBITHA', 'BANGALORE', 'TANUJA', 'KOLKATA', 4.75, 'PENDING', 'TRK100004', '2025-06-25'),
    -> (105, 'RITHIKA', 'HYDERABAD', 'MEGHANA', 'CHENNAI', 3.00, 'DELIVERED', 'TRK100005', '2025-06-10');
Query OK, 5 rows affected (0.03 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM COURIER;
+-----------+------------+---------------+-------------+-----------------+--------+------------+----------------+--------------+
| COURIERID | SENDERNAME | SENDERADDRESS | RECEIVERNAME | RECEIVERADDRESS | WEIGHT | STATUS     | TRACKINGNUMBER | DELIVERYDATE |
+-----------+------------+---------------+-------------+-----------------+--------+------------+----------------+--------------+
|       101 | ELAKKIYA   | CHENNAI       | ROJA        | DELHI           |   3.25 | IN TRANSIT | TRK100001      | 2025-06-18   |
|       102 | LAVANYA    | MADURAI       | SEREESHA    | MUMBAI          |   2.10 | DELIVERED  | TRK100002      | 2025-06-08   |
|       103 | KASHIFA    | COIMBATORE    | HARITHA     | PUNE            |   1.80 | IN TRANSIT | TRK100003      | 2025-06-20   |
|       104 | SHOBITHA   | BANGALORE     | TANUJA      | KOLKATA         |   4.75 | PENDING    | TRK100004      | 2025-06-25   |
|       105 | RITHIKA    | HYDERABAD     | MEGHANA     | CHENNAI         |   3.00 | DELIVERED  | TRK100005      | 2025-06-10   |
+-----------+------------+---------------+-------------+-----------------+--------+------------+----------------+--------------+
5 rows in set (0.00 sec)
```

```
mysql> INSERT INTO COURIER_SERVICES VALUES
    -> (1, 'STANDARD DELIVERY', 150.00),
    -> (2, 'EXPRESS DELIVERY', 250.00),
    -> (3, 'SAME DAY DELIVERY', 350.00);
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM COURIER_SERVICES;
+-----------+------------------+--------+
| SERVICEID | SERVICENAME      | COST   |
+-----------+------------------+--------+
|         1 | STANDARD DELIVERY | 150.00 |
|         2 | EXPRESS DELIVERY  | 250.00 |
|         3 | SAME DAY DELIVERY | 350.00 |
+-----------+------------------+--------+
3 rows in set (0.00 sec)
```

```
mysql> INSERT INTO EMPLOYEE VALUES
    -> (1, 'KARTHIK', 'KARTHIK@CMS.COM', '9000011111', 'DELIVERY AGENT', 22000.00),
    -> (2, 'PRIYA', 'PRIYA@CMS.COM', '9000011112', 'MANAGER', 50000.00),
    -> (3, 'ARUN', 'ARUN@CMS.COM', '9000011113', 'SUPERVISOR', 35000.00),
    -> (4, 'DEVI', 'DEVI@CMS.COM', '9000011114', 'ADMIN', 27000.00),
    -> (5, 'VIKRAM', 'VIKRAM@CMS.COM', '9000011115', 'DELIVERY AGENT', 24000.00);
Query OK, 5 rows affected (0.03 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM EMPLOYEE;
+------------+---------+-----------------+---------------+----------------+----------+
| EMPLOYEEID | NAME    | EMAIL           | CONTACTNUMBER | ROLE           | SALARY   |
+------------+---------+-----------------+---------------+----------------+----------+
|          1 | KARTHIK | KARTHIK@CMS.COM | 9000011111    | DELIVERY AGENT | 22000.00 |
|          2 | PRIYA   | PRIYA@CMS.COM   | 9000011112    | MANAGER        | 50000.00 |
|          3 | ARUN    | ARUN@CMS.COM    | 9000011113    | SUPERVISOR     | 35000.00 |
|          4 | DEVI    | DEVI@CMS.COM    | 9000011114    | ADMIN          | 27000.00 |
|          5 | VIKRAM  | VIKRAM@CMS.COM  | 9000011115    | DELIVERY AGENT | 24000.00 |
+------------+---------+-----------------+---------------+----------------+----------+
5 rows in set (0.00 sec)
```

```
mysql> INSERT INTO LOCATION VALUES
    -> (1, 'CHENNAI HUB', 'CHENNAI, TAMIL NADU'),
    -> (2, 'BANGALORE HUB', 'BANGALORE, KARNATAKA'),
    -> (3, 'DELHI HUB', 'DELHI, INDIA');
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM LOCATION;
+------------+---------------+----------------------+
| LOCATIONID | LOCATIONNAME  | ADDRESS              |
+------------+---------------+----------------------+
|          1 | CHENNAI HUB   | CHENNAI, TAMIL NADU  |
|          2 | BANGALORE HUB | BANGALORE, KARNATAKA |
|          3 | DELHI HUB     | DELHI, INDIA         |
+------------+---------------+----------------------+
3 rows in set (0.00 sec)
```

```
mysql> INSERT INTO PAYMENT VALUES
    -> (1, 101, 1, 150.00, '2025-06-15'),
    -> (2, 102, 3, 250.00, '2025-06-08'),
    -> (3, 103, 2, 150.00, '2025-06-16'),
    -> (4, 104, 2, 250.00, '2025-06-19'),
    -> (5, 105, 1, 350.00, '2025-06-10');
Query OK, 5 rows affected (0.02 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM PAYMENT;
+-----------+-----------+------------+--------+-------------+
| PAYMENTID | COURIERID | LOCATIONID | AMOUNT | PAYMENTDATE |
+-----------+-----------+------------+--------+-------------+
|         1 |       101 |          1 | 150.00 | 2025-06-15  |
|         2 |       102 |          3 | 250.00 | 2025-06-08  |
|         3 |       103 |          2 | 150.00 | 2025-06-16  |
|         4 |       104 |          2 | 250.00 | 2025-06-19  |
|         5 |       105 |          1 | 350.00 | 2025-06-10  |
+-----------+-----------+------------+--------+-------------+
5 rows in set (0.00 sec)
```

Task 2:

Select, Where Solve the following queries in the Schema that you have created above

1. List all customers:

```
mysql> SELECT * FROM USER;
+--------+----------+--------------------+----------+---------------+------------+
| USERID | NAME     | EMAIL              | PASSWORD | CONTACTNUMBER | ADDRESS    |
+--------+----------+--------------------+----------+---------------+------------+
|      1 | ELAKKIYA | ELAKKIYA@GMAIL.COM | PASS123  | 9000000001    | CHENNAI    |
|      2 | LAVANYA  | LAVANYA@GMAIL.COM  | PASS234  | 9000000002    | MADURAI    |
|      3 | KASHIFA  | KASHIFA@GMAIL.COM  | PASS345  | 9000000003    | COIMBATORE |
|      4 | SHOBITHA | SHOBITHA@GMAIL.COM | PASS456  | 9000000004    | BANGALORE  |
|      5 | RITHIKA  | RITHIKA@GMAIL.COM  | PASS567  | 9000000005    | HYDERABAD  |
+--------+----------+--------------------+----------+---------------+------------+
5 rows in set (0.00 sec)
```

2. List all orders for a specific customer:

```
mysql> SELECT * FROM COURIER WHERE SENDERNAME = 'ELAKKIYA';
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+
| COURIERID | SENDERNAME | SENDERADDRESS | RECEIVERNAME | RECEIVERADDRESS | WEIGHT | STATUS     | TRACKINGNUMBER | DELIVERYDATE |
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+
|       101 | ELAKKIYA   | CHENNAI       | ROJA         | DELHI           |   3.25 | IN TRANSIT | TRK100001      | 2025-06-18   |
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+
1 row in set (0.00 sec)
```

3. List all couriers:

```
mysql> SELECT * FROM COURIER;
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+
| COURIERID | SENDERNAME | SENDERADDRESS | RECEIVERNAME | RECEIVERADDRESS | WEIGHT | STATUS     | TRACKINGNUMBER | DELIVERYDATE |
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+
|       101 | ELAKKIYA   | CHENNAI       | ROJA         | DELHI           |   3.25 | IN TRANSIT | TRK100001      | 2025-06-18   |
|       102 | LAVANYA    | MADURAI       | SEREESHA     | MUMBAI          |   2.10 | DELIVERED  | TRK100002      | 2025-06-08   |
|       103 | KASHIFA    | COIMBATORE    | HARITHA      | PUNE            |   1.80 | IN TRANSIT | TRK100003      | 2025-06-20   |
|       104 | SHOBITHA   | BANGALORE     | TANUJA       | KOLKATA         |   4.75 | PENDING    | TRK100004      | 2025-06-25   |
|       105 | RITHIKA    | HYDERABAD     | MEGHANA      | CHENNAI         |   3.00 | DELIVERED  | TRK100005      | 2025-06-10   |
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+
5 rows in set (0.00 sec)
```

4. List all packages for a specific order:

```
mysql> SELECT * FROM COURIER WHERE COURIERID = 102;
+-----------+------------+---------------+--------------+-----------------+--------+-----------+----------------+--------------+
| COURIERID | SENDERNAME | SENDERADDRESS | RECEIVERNAME | RECEIVERADDRESS | WEIGHT | STATUS    | TRACKINGNUMBER | DELIVERYDATE |
+-----------+------------+---------------+--------------+-----------------+--------+-----------+----------------+--------------+
|       102 | LAVANYA    | MADURAI       | SEREESHA     | MUMBAI          |   2.10 | DELIVERED | TRK100002      | 2025-06-08   |
+-----------+------------+---------------+--------------+-----------------+--------+-----------+----------------+--------------+
1 row in set (0.00 sec)
```

5. List all deliveries for a specific courier:

```
mysql>  SELECT * FROM COURIER WHERE COURIERID = 104;
+-----------+------------+---------------+--------------+-----------------+--------+---------+----------------+--------------+
| COURIERID | SENDERNAME | SENDERADDRESS | RECEIVERNAME | RECEIVERADDRESS | WEIGHT | STATUS  | TRACKINGNUMBER | DELIVERYDATE |
+-----------+------------+---------------+--------------+-----------------+--------+---------+----------------+--------------+
|       104 | SHOBITHA   | BANGALORE     | TANUJA       | KOLKATA         |   4.75 | PENDING | TRK100004      | 2025-06-25   |
+-----------+------------+---------------+--------------+-----------------+--------+---------+----------------+--------------+
1 row in set (0.00 sec)
```

6. List all undelivered packages:

```
mysql> SELECT * FROM COURIER WHERE STATUS != 'DELIVERED';
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+
| COURIERID | SENDERNAME | SENDERADDRESS | RECEIVERNAME | RECEIVERADDRESS | WEIGHT | STATUS     | TRACKINGNUMBER | DELIVERYDATE |
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+
|       101 | ELAKKIYA   | CHENNAI       | ROJA         | DELHI           |   3.25 | IN TRANSIT | TRK100001      | 2025-06-18   |
|       103 | KASHIFA    | COIMBATORE    | HARITHA      | PUNE            |   1.80 | IN TRANSIT | TRK100003      | 2025-06-20   |
|       104 | SHOBITHA   | BANGALORE     | TANUJA       | KOLKATA         |   4.75 | PENDING    | TRK100004      | 2025-06-25   |
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+
3 rows in set (0.00 sec)
```

7. List all packages that are scheduled for delivery today:

```
mysql> SELECT * FROM COURIER WHERE DELIVERYDATE = '2025-06-10';
+-----------+------------+---------------+--------------+-----------------+--------+-----------+----------------+--------------+
| COURIERID | SENDERNAME | SENDERADDRESS | RECEIVERNAME | RECEIVERADDRESS | WEIGHT | STATUS    | TRACKINGNUMBER | DELIVERYDATE |
+-----------+------------+---------------+--------------+-----------------+--------+-----------+----------------+--------------+
|       105 | RITHIKA    | HYDERABAD     | MEGHANA      | CHENNAI         |   3.00 | DELIVERED | TRK100005      | 2025-06-10   |
+-----------+------------+---------------+--------------+-----------------+--------+-----------+----------------+--------------+
1 row in set (0.00 sec)
```

8. List all packages with a specific status:

```
mysql> SELECT * FROM COURIER WHERE STATUS = 'IN TRANSIT';
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+
| COURIERID | SENDERNAME | SENDERADDRESS | RECEIVERNAME | RECEIVERADDRESS | WEIGHT | STATUS     | TRACKINGNUMBER | DELIVERYDATE |
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+
|       101 | ELAKKIYA   | CHENNAI       | ROJA         | DELHI           |   3.25 | IN TRANSIT | TRK100001      | 2025-06-18   |
|       103 | KASHIFA    | COIMBATORE    | HARITHA      | PUNE            |   1.80 | IN TRANSIT | TRK100003      | 2025-06-20   |
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+
2 rows in set (0.00 sec)
```

9. Calculate the total number of packages for each courier.

```
mysql> SELECT SENDERNAME, COUNT(*) AS TOTAL_PACKAGES
    -> FROM COURIER
    -> GROUP BY SENDERNAME;
+------------+----------------+
| SENDERNAME | TOTAL_PACKAGES |
+------------+----------------+
| ELAKKIYA   |              1 |
| LAVANYA    |              1 |
| KASHIFA    |              1 |
| SHOBITHA   |              1 |
| RITHIKA    |              1 |
+------------+----------------+
5 rows in set (0.01 sec)
```

10. Find the average delivery time for each courier

```
mysql> SELECT SENDERNAME, ROUND(AVG(DATEDIFF(DELIVERYDATE, '2025-06-05'))) AS AVG_DELIVERY_DAYS FROM COURIER
    -> GROUP BY SENDERNAME;
+------------+--------------------+
| SENDERNAME | AVG_DELIVERY_DAYS  |
+------------+--------------------+
| ELAKKIYA   |                 13 |
| LAVANYA    |                  3 |
| KASHIFA    |                 15 |
| SHOBITHA   |                 20 |
| RITHIKA    |                  5 |
+------------+--------------------+
5 rows in set (0.04 sec)
```

11. List all packages with a specific weight range:

```
mysql> SELECT * FROM COURIER WHERE WEIGHT BETWEEN 2 AND 4;
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+
| COURIERID | SENDERNAME | SENDERADDRESS | RECEIVERNAME | RECEIVERADDRESS | WEIGHT | STATUS     | TRACKINGNUMBER | DELIVERYDATE |
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+
|       101 | ELAKKIYA   | CHENNAI       | ROJA         | DELHI           |   3.25 | IN TRANSIT | TRK100001      | 2025-06-18   |
|       102 | LAVANYA    | MADURAI       | SEREESHA     | MUMBAI          |   2.10 | DELIVERED  | TRK100002      | 2025-06-08   |
|       105 | RITHIKA    | HYDERABAD     | MEGHANA      | CHENNAI         |   3.00 | DELIVERED  | TRK100005      | 2025-06-10   |
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+
3 rows in set (0.00 sec)
```

12. Retrieve employees whose names contain 'John'

```
mysql> SELECT * FROM EMPLOYEE WHERE NAME = 'JOHN';
Empty set (0.01 sec)

mysql> SELECT * FROM EMPLOYEE WHERE NAME LIKE 'JOHN';
Empty set (0.00 sec)
```

13. Retrieve all courier records with payments greater than 150.

```
mysql> SELECT C.COURIERID, C.SENDERNAME, P.AMOUNT FROM COURIER C, PAYMENT P WHERE C.COURIERID = P.COURIERID AND P.AMOUNT > 150;
+-----------+------------+--------+
| COURIERID | SENDERNAME | AMOUNT |
+-----------+------------+--------+
|       102 | LAVANYA    | 250.00 |
|       104 | SHOBITHA   | 250.00 |
|       105 | RITHIKA    | 350.00 |
+-----------+------------+--------+
3 rows in set (0.00 sec)
```

Task 3:

GroupBy, Aggregate Functions, Having, Order By, where

14. Find the total number of couriers handled by each employee.

```
mysql> SELECT E.EMPLOYEEID, E.NAME, COUNT(C.COURIERID) AS TOTAL_COURIER FROM EMPLOYEE E
    -> LEFT JOIN COURIER C ON E.EMPLOYEEID = C.EMPLOYEEID
    -> GROUP BY E.EMPLOYEEID, E.NAME;
+------------+---------+---------------+
| EMPLOYEEID | NAME    | TOTAL_COURIER |
+------------+---------+---------------+
|          1 | KARTHIK |             1 |
|          2 | PRIYA   |             1 |
|          3 | ARUN    |             1 |
|          4 | DEVI    |             1 |
|          5 | VIKRAM  |             1 |
+------------+---------+---------------+
5 rows in set (0.02 sec)
```

15. Calculate the total revenue generated by each location

```
mysql> SELECT LOCATIONID, SUM(AMOUNT) AS TOTAL_REVENUE
    -> FROM PAYMENT
    -> GROUP BY LOCATIONID;
+------------+---------------+
| LOCATIONID | TOTAL_REVENUE |
+------------+---------------+
|          1 |        500.00 |
|          2 |        400.00 |
|          3 |        250.00 |
+------------+---------------+
3 rows in set (0.00 sec)
```

16. Find the total number of couriers delivered to each location.

```
mysql> SELECT LOCATIONID, COUNT(*) AS TOTAL_COURIERS
    -> FROM PAYMENT
    -> GROUP BY LOCATIONID;
+------------+-----------------+
| LOCATIONID | TOTAL_COURIERS |
+------------+-----------------+
|          1 |               2 |
|          2 |               2 |
|          3 |               1 |
+------------+-----------------+
3 rows in set (0.00 sec)
```

17. Find the courier with the highest average delivery time:

```
mysql> SELECT COURIERID,ROUND( AVG(DATEDIFF(DELIVERYDATE, '2025-06-05' ))) AS AVG_DAYS
    -> FROM COURIER
    -> GROUP BY COURIERID
    -> ORDER BY AVG_DAYS DESC
    -> LIMIT 1;
+-----------+----------+
| COURIERID | AVG_DAYS |
+-----------+----------+
|       104 |       20 |
+-----------+----------+
1 row in set (0.00 sec)
```

18. Find Locations with Total Payments Less Than a Certain Amount

```
mysql> SELECT LOCATIONID, SUM(AMOUNT) AS TOTAL
    -> FROM PAYMENT
    -> GROUP BY LOCATIONID HAVING TOTAL < 500;
+------------+--------+
| LOCATIONID | TOTAL  |
+------------+--------+
|          2 | 400.00 |
|          3 | 250.00 |
+------------+--------+
2 rows in set (0.00 sec)
```

19. Calculate Total Payments per Location

```
mysql> SELECT LOCATIONID, SUM(AMOUNT) AS TOTAL
    -> FROM PAYMENT
    -> GROUP BY LOCATIONID;
+------------+--------+
| LOCATIONID | TOTAL  |
+------------+--------+
|          1 | 500.00 |
|          2 | 400.00 |
|          3 | 250.00 |
+------------+--------+
3 rows in set (0.00 sec)
```

20. Retrieve couriers who have received payments totaling more than 300 in a specific location (LocationID = X):

```
mysql> SELECT COURIERID, SUM(AMOUNT) AS TOTAL FROM PAYMENT
    -> WHERE LOCATIONID = 1
    -> GROUP BY COURIERID HAVING TOTAL >300;
+-----------+--------+
| COURIERID | TOTAL  |
+-----------+--------+
|       105 | 350.00 |
+-----------+--------+
1 row in set (0.00 sec)
```

21. Retrieve couriers who have received payments totaling more than 100 after a certain date (PaymentDate > 'YYYY-MM-DD'):

```
mysql> SELECT COURIERID, SUM(AMOUNT) AS TOTAL FROM PAYMENT WHERE PAYMENTDATE > '2025-06-10'
    -> GROUP BY COURIERID HAVING TOTAL >100;
+-----------+--------+
| COURIERID | TOTAL  |
+-----------+--------+
|       101 | 150.00 |
|       103 | 150.00 |
|       104 | 250.00 |
+-----------+--------+
3 rows in set (0.00 sec)
```

22. Retrieve locations where the total amount received is more than 200 before a certain date (PaymentDate > 'YYYY-MM-DD')

```
mysql> SELECT LOCATIONID, SUM(AMOUNT) AS TOTAL FROM PAYMENT
    -> WHERE PAYMENTDATE < '2025-06-16'
    -> GROUP BY LOCATIONID HAVING TOTAL > 200;
+------------+--------+
| LOCATIONID | TOTAL  |
+------------+--------+
|          1 | 500.00 |
|          3 | 250.00 |
+------------+--------+
2 rows in set (0.00 sec)
```

```
mysql> SELECT L.LOCATIONID, L.LOCATIONNAME, SUM(AMOUNT) AS TOTAL FROM PAYMENT P
    -> JOIN LOCATION L ON P.LOCATIONID = L.LOCATIONID
    -> WHERE P.PAYMENTDATE < '2025-06-16'
    -> GROUP BY L.LOCATIONID,L.LOCATIONNAME HAVING TOTAL >200;
+------------+--------------+--------+
| LOCATIONID | LOCATIONNAME | TOTAL  |
+------------+--------------+--------+
|          1 | CHENNAI HUB  | 500.00 |
|          3 | DELHI HUB    | 250.00 |
+------------+--------------+--------+
2 rows in set (0.01 sec)
```

Task 4:

Inner Join, Full Outer Join, Cross Join, Left Outer Join, Right Outer Join

23. Retrieve Payments with Courier Information

```
mysql> SELECT P.*, C.SENDERNAME, C.RECEIVERNAME, C.STATUS
    -> FROM PAYMENT P
    -> INNER JOIN COURIER C ON P.COURIERID = C.COURIERID;
+-----------+-----------+------------+--------+-------------+------------+--------------+------------+
| PAYMENTID | COURIERID | LOCATIONID | AMOUNT | PAYMENTDATE | SENDERNAME | RECEIVERNAME | STATUS     |
+-----------+-----------+------------+--------+-------------+------------+--------------+------------+
|         1 |       101 |          1 | 150.00 | 2025-06-15  | ELAKKIYA   | ROJA         | IN TRANSIT |
|         2 |       102 |          3 | 250.00 | 2025-06-08  | LAVANYA    | SEREESHA     | DELIVERED  |
|         3 |       103 |          2 | 150.00 | 2025-06-16  | KASHIFA    | HARITHA      | IN TRANSIT |
|         4 |       104 |          2 | 250.00 | 2025-06-19  | SHOBITHA   | TANUJA       | PENDING    |
|         5 |       105 |          1 | 350.00 | 2025-06-10  | RITHIKA    | MEGHANA      | DELIVERED  |
+-----------+-----------+------------+--------+-------------+------------+--------------+------------+
5 rows in set (0.01 sec)
```

24. Retrieve Payments with Location Information

```
mysql> SELECT P.*, L.LOCATIONNAME, L.ADDRESS FROM PAYMENT P
    -> INNER JOIN LOCATION L ON P.LOCATIONID = L.LOCATIONID;
+-----------+-----------+------------+--------+-------------+---------------+----------------------+
| PAYMENTID | COURIERID | LOCATIONID | AMOUNT | PAYMENTDATE | LOCATIONNAME  | ADDRESS              |
+-----------+-----------+------------+--------+-------------+---------------+----------------------+
|         1 |       101 |          1 | 150.00 | 2025-06-15  | CHENNAI HUB   | CHENNAI, TAMIL NADU  |
|         5 |       105 |          1 | 350.00 | 2025-06-10  | CHENNAI HUB   | CHENNAI, TAMIL NADU  |
|         3 |       103 |          2 | 150.00 | 2025-06-16  | BANGALORE HUB | BANGALORE, KARNATAKA |
|         4 |       104 |          2 | 250.00 | 2025-06-19  | BANGALORE HUB | BANGALORE, KARNATAKA |
|         2 |       102 |          3 | 250.00 | 2025-06-08  | DELHI HUB     | DELHI, INDIA         |
+-----------+-----------+------------+--------+-------------+---------------+----------------------+
5 rows in set (0.01 sec)
```

25. Retrieve Payments with Courier and Location Information

```
mysql> SELECT P.*, C.SENDERNAME, C.RECEIVERNAME, C.STATUS,L.LOCATIONNAME, L.ADDRESS FROM PAYMENT P
    -> JOIN  COURIER C ON P.COURIERID = C.COURIERID
    -> JOIN LOCATION L ON P.LOCATIONID = L.LOCATIONID;
+-----------+-----------+------------+--------+-------------+------------+--------------+------------+---------------+----------------------+
| PAYMENTID | COURIERID | LOCATIONID | AMOUNT | PAYMENTDATE | SENDERNAME | RECEIVERNAME | STATUS     | LOCATIONNAME  | ADDRESS              |
+-----------+-----------+------------+--------+-------------+------------+--------------+------------+---------------+----------------------+
|         1 |       101 |          1 | 150.00 | 2025-06-15  | ELAKKIYA   | ROJA         | IN TRANSIT | CHENNAI HUB   | CHENNAI, TAMIL NADU  |
|         5 |       105 |          1 | 350.00 | 2025-06-10  | RITHIKA    | MEGHANA      | DELIVERED  | CHENNAI HUB   | CHENNAI, TAMIL NADU  |
|         3 |       103 |          2 | 150.00 | 2025-06-16  | KASHIFA    | HARITHA      | IN TRANSIT | BANGALORE HUB | BANGALORE, KARNATAKA |
|         4 |       104 |          2 | 250.00 | 2025-06-19  | SHOBITHA   | TANUJA       | PENDING    | BANGALORE HUB | BANGALORE, KARNATAKA |
|         2 |       102 |          3 | 250.00 | 2025-06-08  | LAVANYA    | SEREESHA     | DELIVERED  | DELHI HUB     | DELHI, INDIA         |
+-----------+-----------+------------+--------+-------------+------------+--------------+------------+---------------+----------------------+
5 rows in set (0.00 sec)
```

26. List all payments with courier details

```
mysql> SELECT P.*, C.SENDERNAME, C.RECEIVERNAME, C.STATUS
    -> FROM PAYMENT P
    -> INNER JOIN COURIER C ON P.COURIERID = C.COURIERID;
+-----------+-----------+------------+--------+-------------+------------+--------------+------------+
| PAYMENTID | COURIERID | LOCATIONID | AMOUNT | PAYMENTDATE | SENDERNAME | RECEIVERNAME | STATUS     |
+-----------+-----------+------------+--------+-------------+------------+--------------+------------+
|         1 |       101 |          1 | 150.00 | 2025-06-15  | ELAKKIYA   | ROJA         | IN TRANSIT |
|         2 |       102 |          3 | 250.00 | 2025-06-08  | LAVANYA    | SEREESHA     | DELIVERED  |
|         3 |       103 |          2 | 150.00 | 2025-06-16  | KASHIFA    | HARITHA      | IN TRANSIT |
|         4 |       104 |          2 | 250.00 | 2025-06-19  | SHOBITHA   | TANUJA       | PENDING    |
|         5 |       105 |          1 | 350.00 | 2025-06-10  | RITHIKA    | MEGHANA      | DELIVERED  |
+-----------+-----------+------------+--------+-------------+------------+--------------+------------+
5 rows in set (0.01 sec)
```

27. Total payments received for each courier

```
mysql> SELECT COURIERID, SUM(AMOUNT) AS TOTAL_PAYMENT
    -> FROM PAYMENT
    -> GROUP BY COURIERID;
+-----------+---------------+
| COURIERID | TOTAL_PAYMENT |
+-----------+---------------+
|       101 |        150.00 |
|       102 |        250.00 |
|       103 |        150.00 |
|       104 |        250.00 |
|       105 |        350.00 |
+-----------+---------------+
5 rows in set (0.03 sec)
```

28. List payments made on a specific date

```
mysql> SELECT * FROM PAYMENT WHERE PAYMENTDATE = '2025-06-10';
+-----------+-----------+------------+--------+-------------+
| PAYMENTID | COURIERID | LOCATIONID | AMOUNT | PAYMENTDATE |
+-----------+-----------+------------+--------+-------------+
|         5 |       105 |          1 | 350.00 | 2025-06-10  |
+-----------+-----------+------------+--------+-------------+
1 row in set (0.01 sec)
```

29. Get Courier Information for Each Payment

```
mysql> SELECT P.PAYMENTID, P.AMOUNT, C.* FROM PAYMENT P
    -> JOIN COURIER C ON P.COURIERID = C.COURIERID;
+-----------+--------+-----------+------------+---------------+-------------+-----------------+--------+-----------+----------------+--------------+
| PAYMENTID | AMOUNT | COURIERID | SENDERNAME | SENDERADDRESS | RECEIVERNAME | RECEIVERADDRESS | WEIGHT | STATUS    | TRACKINGNUMBER | DELIVERYDATE |
+-----------+--------+-----------+------------+---------------+-------------+-----------------+--------+-----------+----------------+--------------+
|         1 | 150.00 |       101 | ELAKKIYA   | CHENNAI       | ROJA        | DELHI           |   3.25 | IN TRANSIT | TRK100001      | 2025-06-18   |
|         2 | 250.00 |       102 | LAVANYA    | MADURAI       | SEREESHA    | MUMBAI          |   2.10 | DELIVERED  | TRK100002      | 2025-06-08   |
|         3 | 150.00 |       103 | KASHIFA    | COIMBATORE    | HARITHA     | PUNE            |   1.80 | IN TRANSIT | TRK100003      | 2025-06-20   |
|         4 | 250.00 |       104 | SHOBITHA   | BANGALORE     | TANUJA      | KOLKATA         |   4.75 | PENDING    | TRK100004      | 2025-06-25   |
|         5 | 350.00 |       105 | RITHIKA    | HYDERABAD     | MEGHANA     | CHENNAI         |   3.00 | DELIVERED  | TRK100005      | 2025-06-10   |
+-----------+--------+-----------+------------+---------------+-------------+-----------------+--------+-----------+----------------+--------------+
5 rows in set (0.00 sec)
```

30. Get Payment Details with Location

```
mysql> SELECT P.PAYMENTID, P.AMOUNT, P.PAYMENTDATE, L.LOCATIONNAME
    -> FROM PAYMENT P
    -> JOIN LOCATION L ON P.LOCATIONID = L.LOCATIONID;
+-----------+--------+-------------+---------------+
| PAYMENTID | AMOUNT | PAYMENTDATE | LOCATIONNAME  |
+-----------+--------+-------------+---------------+
|         1 | 150.00 | 2025-06-15  | CHENNAI HUB   |
|         5 | 350.00 | 2025-06-10  | CHENNAI HUB   |
|         3 | 150.00 | 2025-06-16  | BANGALORE HUB |
|         4 | 250.00 | 2025-06-19  | BANGALORE HUB |
|         2 | 250.00 | 2025-06-08  | DELHI HUB     |
+-----------+--------+-------------+---------------+
5 rows in set (0.00 sec)
```

31. Calculating Total Payments for Each Courier

```
mysql> SELECT C.COURIERID, C.TRACKINGNUMBER, SUM(P.AMOUNT) AS TOTAL_PAYMENT
    -> FROM COURIER C
    -> JOIN PAYMENT P ON C.COURIERID = P.COURIERID
    -> GROUP BY C.COURIERID, C.TRACKINGNUMBER;
+-----------+----------------+---------------+
| COURIERID | TRACKINGNUMBER | TOTAL_PAYMENT |
+-----------+----------------+---------------+
|       101 | TRK100001      |        150.00 |
|       102 | TRK100002      |        250.00 |
|       103 | TRK100003      |        150.00 |
|       104 | TRK100004      |        250.00 |
|       105 | TRK100005      |        350.00 |
+-----------+----------------+---------------+
5 rows in set (0.02 sec)
```

32. List Payments Within a Date Range

```
mysql> SELECT * FROM PAYMENT WHERE PAYMENTDATE BETWEEN '2025-06-04' AND '2025-06-15';
+-----------+-----------+------------+--------+-------------+
| PAYMENTID | COURIERID | LOCATIONID | AMOUNT | PAYMENTDATE |
+-----------+-----------+------------+--------+-------------+
|         1 |       101 |          1 | 150.00 | 2025-06-15  |
|         2 |       102 |          3 | 250.00 | 2025-06-08  |
|         5 |       105 |          1 | 350.00 | 2025-06-10  |
+-----------+-----------+------------+--------+-------------+
3 rows in set (0.00 sec)
```

33. Retrieve a list of all users and their corresponding courier records, including cases where there are no matches on either side

```
mysql> SELECT U.USERID, U.NAME AS USERNAME, C.COURIERID, C.TRACKINGNUMBER
    -> FROM USER U
    -> JOIN COURIER C ON U.NAME = C.SENDERNAME;
+--------+----------+-----------+----------------+
| USERID | USERNAME | COURIERID | TRACKINGNUMBER |
+--------+----------+-----------+----------------+
|      1 | ELAKKIYA |       101 | TRK100001      |
|      2 | LAVANYA  |       102 | TRK100002      |
|      3 | KASHIFA  |       103 | TRK100003      |
|      4 | SHOBITHA |       104 | TRK100004      |
|      5 | RITHIKA  |       105 | TRK100005      |
+--------+----------+-----------+----------------+
5 rows in set (0.02 sec)
```

34. Retrieve a list of all couriers and their corresponding services, including cases where there are no matches on either side

```
mysql> SELECT P.COURIERID, P.AMOUNT, CS.SERVICENAME FROM PAYMENT P
    -> JOIN COURIER_SERVICES CS ON P.AMOUNT = CS.COST;
+-----------+--------+------------------+
| COURIERID | AMOUNT | SERVICENAME      |
+-----------+--------+------------------+
|       101 | 150.00 | STANDARD DELIVERY |
|       102 | 250.00 | EXPRESS DELIVERY  |
|       103 | 150.00 | STANDARD DELIVERY |
|       104 | 250.00 | EXPRESS DELIVERY  |
|       105 | 350.00 | SAME DAY DELIVERY |
+-----------+--------+------------------+
5 rows in set (0.00 sec)

mysql>
```

35. Retrieve a list of all employees and their corresponding payments, including cases where there are no matches on either side

```
mysql> ALTER TABLE COURIER
    -> ADD EMPLOYEEID INT,
    -> ADD FOREIGN KEY (EMPLOYEEID) REFERENCES EMPLOYEE(EMPLOYEEID);
Query OK, 5 rows affected (0.26 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

```
mysql> ALTER TABLE COURIER
    -> ADD SERVICEID INT,
    -> ADD FOREIGN KEY (SERVICEID) REFERENCES COURIER_SERVICES(SERVICEID);
Query OK, 5 rows affected (0.55 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

```
mysql> SELECT * FROM COURIER;
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+-----------+------------+
| COURIERID | SENDERNAME | SENDERADDRESS | RECEIVERNAME | RECEIVERADDRESS | WEIGHT | STATUS     | TRACKINGNUMBER | DELIVERYDATE | SERVICEID | EMPLOYEEID |
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+-----------+------------+
|       101 | ELAKKIYA   | CHENNAI       | ROJA         | DELHI           |   3.25 | IN TRANSIT | TRK100001      | 2025-06-18   |      NULL |       NULL |
|       102 | LAVANYA    | MADURAI       | SEREESHA     | MUMBAI          |   2.10 | DELIVERED  | TRK100002      | 2025-06-08   |      NULL |       NULL |
|       103 | KASHIFA    | COIMBATORE    | HARITHA      | PUNE            |   1.80 | IN TRANSIT | TRK100003      | 2025-06-20   |      NULL |       NULL |
|       104 | SHOBITHA   | BANGALORE     | TANUJA       | KOLKATA         |   4.75 | PENDING    | TRK100004      | 2025-06-25   |      NULL |       NULL |
|       105 | RITHIKA    | HYDERABAD     | MEGHANA      | CHENNAI         |   3.00 | DELIVERED  | TRK100005      | 2025-06-10   |      NULL |       NULL |
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+-----------+------------+
5 rows in set (0.01 sec)
```

```
mysql> UPDATE COURIER
    -> SET EMPLOYEEID = 2, SERVICEID = 1 WHERE COURIERID = 101;
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE COURIER
    -> SET EMPLOYEEID = 1, SERVICEID = 2 WHERE COURIERID = 102;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE COURIER
    -> SET EMPLOYEEID = 3, SERVICEID = 1 WHERE COURIERID = 103;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE COURIER
    -> SET EMPLOYEEID = 4, SERVICEID = 2 WHERE COURIERID = 104;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE COURIER
    -> SET EMPLOYEEID = 5, SERVICEID = 3 WHERE COURIERID = 105;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> SELECT * FROM COURIER;
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+-----------+------------+
| COURIERID | SENDERNAME | SENDERADDRESS | RECEIVERNAME | RECEIVERADDRESS | WEIGHT | STATUS     | TRACKINGNUMBER | DELIVERYDATE | SERVICEID | EMPLOYEEID |
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+-----------+------------+
|       101 | ELAKKIYA   | CHENNAI       | ROJA         | DELHI           |   3.25 | IN TRANSIT | TRK100001      | 2025-06-18   |         1 |          2 |
|       102 | LAVANYA    | MADURAI       | SEREESHA     | MUMBAI          |   2.10 | DELIVERED  | TRK100002      | 2025-06-08   |         2 |          1 |
|       103 | KASHIFA    | COIMBATORE    | HARITHA      | PUNE            |   1.80 | IN TRANSIT | TRK100003      | 2025-06-20   |         1 |          3 |
|       104 | SHOBITHA   | BANGALORE     | TANUJA       | KOLKATA         |   4.75 | PENDING    | TRK100004      | 2025-06-25   |         2 |          4 |
|       105 | RITHIKA    | HYDERABAD     | MEGHANA      | CHENNAI         |   3.00 | DELIVERED  | TRK100005      | 2025-06-10   |         3 |          5 |
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+-----------+------------+
5 rows in set (0.00 sec)
```

```
mysql> SELECT E.EMPLOYEEID, E.NAME, P.AMOUNT
    -> FROM EMPLOYEE E
    -> LEFT JOIN COURIER C ON E.EMPLOYEEID = C.EMPLOYEEID
    -> LEFT JOIN PAYMENT P ON C.COURIERID = P.COURIERID;
+------------+---------+--------+
| EMPLOYEEID | NAME    | AMOUNT |
+------------+---------+--------+
|          1 | KARTHIK | 250.00 |
|          2 | PRIYA   | 150.00 |
|          3 | ARUN    | 150.00 |
|          4 | DEVI    | 250.00 |
|          5 | VIKRAM  | 350.00 |
+------------+---------+--------+
5 rows in set (0.00 sec)
```

36. List all users and all courier services, showing all possible combinations.

```
mysql> SELECT U.NAME , CS.SERVICENAME FROM USER U
    -> CROSS JOIN COURIER_SERVICES CS;
+----------+------------------+
| NAME     | SERVICENAME      |
+----------+------------------+
| ELAKKIYA | SAME DAY DELIVERY |
| ELAKKIYA | EXPRESS DELIVERY  |
| ELAKKIYA | STANDARD DELIVERY |
| LAVANYA  | SAME DAY DELIVERY |
| LAVANYA  | EXPRESS DELIVERY  |
| LAVANYA  | STANDARD DELIVERY |
| KASHIFA  | SAME DAY DELIVERY |
| KASHIFA  | EXPRESS DELIVERY  |
| KASHIFA  | STANDARD DELIVERY |
| SHOBITHA | SAME DAY DELIVERY |
| SHOBITHA | EXPRESS DELIVERY  |
| SHOBITHA | STANDARD DELIVERY |
| RITHIKA  | SAME DAY DELIVERY |
| RITHIKA  | EXPRESS DELIVERY  |
| RITHIKA  | STANDARD DELIVERY |
+----------+------------------+
15 rows in set (0.00 sec)
```

37. List all employees and all locations, showing all possible combinations:

```
mysql> SELECT E.NAME AS EMPLOYEE_NAME, L.LOCATIONNAME FROM EMPLOYEE E
    -> CROSS JOIN LOCATION L;
+---------------+---------------+
| EMPLOYEE_NAME | LOCATIONNAME  |
+---------------+---------------+
| KARTHIK       | DELHI HUB     |
| KARTHIK       | BANGALORE HUB |
| KARTHIK       | CHENNAI HUB   |
| PRIYA         | DELHI HUB     |
| PRIYA         | BANGALORE HUB |
| PRIYA         | CHENNAI HUB   |
| ARUN          | DELHI HUB     |
| ARUN          | BANGALORE HUB |
| ARUN          | CHENNAI HUB   |
| DEVI          | DELHI HUB     |
| DEVI          | BANGALORE HUB |
| DEVI          | CHENNAI HUB   |
| VIKRAM        | DELHI HUB     |
| VIKRAM        | BANGALORE HUB |
| VIKRAM        | CHENNAI HUB   |
+---------------+---------------+
15 rows in set (0.00 sec)
```

38. Retrieve a list of couriers and their corresponding sender information (if available)

```
mysql> SELECT C.COURIERID, C.SENDERNAME, U.USERID, U.EMAIL
    -> FROM COURIER C
    -> LEFT JOIN USER U ON C.SENDERNAME = U.NAME;
+-----------+------------+--------+--------------------+
| COURIERID | SENDERNAME | USERID | EMAIL              |
+-----------+------------+--------+--------------------+
|       101 | ELAKKIYA   |      1 | ELAKKIYA@GMAIL.COM |
|       102 | LAVANYA    |      2 | LAVANYA@GMAIL.COM  |
|       103 | KASHIFA    |      3 | KASHIFA@GMAIL.COM  |
|       104 | SHOBITHA   |      4 | SHOBITHA@GMAIL.COM |
|       105 | RITHIKA    |      5 | RITHIKA@GMAIL.COM  |
+-----------+------------+--------+--------------------+
5 rows in set (0.01 sec)
```

39. Retrieve a list of couriers and their corresponding receiver information (if available):

```
mysql> SELECT C.COURIERID, C.RECEIVERNAME, C.RECEIVERADDRESS FROM COURIER
    -> C
    -> ;
+-----------+-------------+-----------------+
| COURIERID | RECEIVERNAME | RECEIVERADDRESS |
+-----------+-------------+-----------------+
|       101 | ROJA         | DELHI           |
|       102 | SEREESHA     | MUMBAI          |
|       103 | HARITHA      | PUNE            |
|       104 | TANUJA       | KOLKATA         |
|       105 | MEGHANA      | CHENNAI         |
+-----------+-------------+-----------------+
5 rows in set (0.00 sec)
```

40. Retrieve a list of couriers along with the courier service details (if available):

```
mysql> SELECT P.COURIERID, P.AMOUNT, CS.SERVICENAME FROM PAYMENT P
    -> JOIN COURIER_SERVICES CS ON P.AMOUNT = CS.COST;
+-----------+--------+-------------------+
| COURIERID | AMOUNT | SERVICENAME       |
+-----------+--------+-------------------+
|       101 | 150.00 | STANDARD DELIVERY |
|       102 | 250.00 | EXPRESS DELIVERY  |
|       103 | 150.00 | STANDARD DELIVERY |
|       104 | 250.00 | EXPRESS DELIVERY  |
|       105 | 350.00 | SAME DAY DELIVERY |
+-----------+--------+-------------------+
5 rows in set (0.00 sec)
```

41. Retrieve a list of employees and the number of couriers assigned to each employee:

```
mysql> SELECT E.EMPLOYEEID, E.NAME, COUNT(C.COURIERID) AS TOTAL_COURIER FROM EMPLOYEE E
    -> LEFT JOIN COURIER C ON E.EMPLOYEEID = C.EMPLOYEEID
    -> GROUP BY E.EMPLOYEEID, E.NAME;
+------------+---------+---------------+
| EMPLOYEEID | NAME    | TOTAL_COURIER |
+------------+---------+---------------+
|          1 | KARTHIK |             1 |
|          2 | PRIYA   |             1 |
|          3 | ARUN    |             1 |
|          4 | DEVI    |             1 |
|          5 | VIKRAM  |             1 |
+------------+---------+---------------+
5 rows in set (0.02 sec)
```

42. Retrieve a list of locations and the total payment amount received at each location:

```
mysql> SELECT L.LOCATIONNAME, SUM(P.AMOUNT) AS TOTAL_AMOUNT FROM LOCATION L
    -> JOIN PAYMENT P ON L.LOCATIONID = P.LOCATIONID
    -> GROUP BY L.LOCATIONNAME;
+---------------+--------------+
| LOCATIONNAME  | TOTAL_AMOUNT |
+---------------+--------------+
| CHENNAI HUB   |       500.00 |
| BANGALORE HUB |       400.00 |
| DELHI HUB     |       250.00 |
+---------------+--------------+
3 rows in set (0.00 sec)
```

43. Retrieve all couriers sent by the same sender (based on SenderName).

```
mysql> SELECT SENDERNAME, COUNT(*) AS TOTAL_COURIERS
    -> FROM COURIER
    -> GROUP BY SENDERNAME HAVING TOTAL_COURIERS > 1;
Empty set (0.00 sec)
```

44. List all employees who share the same role.

```
mysql> SELECT * FROM EMPLOYEE WHERE ROLE IN (
    -> SELECT ROLE FROM EMPLOYEE GROUP BY ROLE HAVING COUNT(*)>1);
+------------+---------+-----------------+---------------+----------------+----------+
| EMPLOYEEID | NAME    | EMAIL           | CONTACTNUMBER | ROLE           | SALARY   |
+------------+---------+-----------------+---------------+----------------+----------+
|          1 | KARTHIK | KARTHIK@CMS.COM | 9000011111    | DELIVERY AGENT | 22000.00 |
|          5 | VIKRAM  | VIKRAM@CMS.COM  | 9000011115    | DELIVERY AGENT | 24000.00 |
+------------+---------+-----------------+---------------+----------------+----------+
2 rows in set (0.00 sec)
```

45. Retrieve all payments made for couriers sent from the same location.

```
mysql> SELECT P.* FROM PAYMENT P
    -> JOIN COURIER C ON P.COURIERID = C.COURIERID
    -> WHERE C.SENDERADDRESS IN (SELECT SENDERADDRESS FROM COURIER GROUP BY SENDERADDRESS HAVING COUNT(*) > 1);
Empty set (0.00 sec)
```

46. Retrieve all couriers sent from the same location (based on SenderAddress).

```
mysql> SELECT SENDERADDRESS , COUNT(*) AS TOTAL_COURIERS
    -> FROM COURIER
    -> GROUP BY SENDERADDRESS HAVING TOTAL_COURIERS > 1;
Empty set (0.00 sec)
```

Same as Q 45 Each sender has sent one courier and each one is sent from different location so they return empty set.

47. List employees and the number of couriers they have delivered:

```
mysql> SELECT E.EMPLOYEEID, E.NAME, COUNT(C.COURIERID) AS TOTAL_COURIER FROM EMPLOYEE E
    -> LEFT JOIN COURIER C ON E.EMPLOYEEID = C.EMPLOYEEID
    -> GROUP BY E.EMPLOYEEID, E.NAME;
+------------+---------+---------------+
| EMPLOYEEID | NAME    | TOTAL_COURIER |
+------------+---------+---------------+
|          1 | KARTHIK |             1 |
|          2 | PRIYA   |             1 |
|          3 | ARUN    |             1 |
|          4 | DEVI    |             1 |
|          5 | VIKRAM  |             1 |
+------------+---------+---------------+
5 rows in set (0.02 sec)
```

48. Find couriers that were paid an amount greater than the cost of their respective courier services

    Nil

```
mysql> SELECT C.COURIERID, P.AMOUNT AS PAID, CS.COST AS SERVICE_COST FROM COURIER C
    -> JOIN PAYMENT P ON C.COURIERID = P.COURIERID
    -> JOIN COURIER_SERVICES CS ON C.SERVICEID = CS.SERVICEID
    -> WHERE P.AMOUNT > CS.COST;
Empty set (0.01 sec)
```

Scope: Inner Queries, Non Equi Joins, Equi joins,Exist,Any,All

49. Find couriers that have a weight greater than the average weight of all couriers

```
mysql> SELECT * FROM COURIER WHERE WEIGHT > (SELECT AVG(WEIGHT) FROM COURIER);
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+
| COURIERID | SENDERNAME | SENDERADDRESS | RECEIVERNAME | RECEIVERADDRESS | WEIGHT | STATUS     | TRACKINGNUMBER | DELIVERYDATE |
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+
|       101 | ELAKKIYA   | CHENNAI       | ROJA         | DELHI           |   3.25 | IN TRANSIT | TRK100001      | 2025-06-18   |
|       104 | SHOBITHA   | BANGALORE     | TANUJA       | KOLKATA         |   4.75 | PENDING    | TRK100004      | 2025-06-25   |
|       105 | RITHIKA    | HYDERABAD     | MEGHANA      | CHENNAI         |   3.00 | DELIVERED  | TRK100005      | 2025-06-10   |
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+
3 rows in set (0.00 sec)
```

50. Find the names of all employees who have a salary greater than the average salary:

```
mysql> SELECT * FROM EMPLOYEE WHERE SALARY > (SELECT AVG(SALARY) FROM EMPLOYEE);
+-------------+-------+---------------+---------------+------------+----------+
| EMPLOYEEID | NAME   | EMAIL         | CONTACTNUMBER | ROLE       | SALARY   |
+-------------+-------+---------------+---------------+------------+----------+
|           2 | PRIYA | PRIYA@CMS.COM | 9000011112    | MANAGER    | 50000.00 |
|           3 | ARUN  | ARUN@CMS.COM  | 9000011113    | SUPERVISOR | 35000.00 |
+-------------+-------+---------------+---------------+------------+----------+
2 rows in set (0.00 sec)
```

51. Find the total cost of all courier services where the cost is less than the maximum cost.

```
mysql> SELECT SUM(COST) AS TOTAL_COST FROM COURIER_SERVICES
    -> WHERE COST < (SELECT MAX(COST) FROM COURIER_SERVICES);
+------------+
| TOTAL_COST |
+------------+
|     400.00 |
+------------+
1 row in set (0.00 sec)
```

52. Find all couriers that have been paid for

```
mysql> SELECT * FROM COURIER
    -> WHERE COURIERID IN ( SELECT COURIERID FROM PAYMENT);
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+
| COURIERID | SENDERNAME | SENDERADDRESS | RECEIVERNAME | RECEIVERADDRESS | WEIGHT | STATUS     | TRACKINGNUMBER | DELIVERYDATE |
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+
|       101 | ELAKKIYA   | CHENNAI       | ROJA         | DELHI           |   3.25 | IN TRANSIT | TRK100001      | 2025-06-18   |
|       102 | LAVANYA    | MADURAI       | SEREESHA     | MUMBAI          |   2.10 | DELIVERED  | TRK100002      | 2025-06-08   |
|       103 | KASHIFA    | COIMBATORE    | HARITHA      | PUNE            |   1.80 | IN TRANSIT | TRK100003      | 2025-06-20   |
|       104 | SHOBITHA   | BANGALORE     | TANUJA       | KOLKATA         |   4.75 | PENDING    | TRK100004      | 2025-06-25   |
|       105 | RITHIKA    | HYDERABAD     | MEGHANA      | CHENNAI         |   3.00 | DELIVERED  | TRK100005      | 2025-06-10   |
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+
5 rows in set (0.00 sec)
```

53. Find the locations where the maximum payment amount was made

```
mysql> SELECT L.LOCATIONNAME, P.AMOUNT FROM PAYMENT P
    -> JOIN LOCATION L ON P.LOCATIONID = L.LOCATIONID
    -> WHERE P.AMOUNT = (SELECT MAX(AMOUNT) FROM PAYMENT);
+--------------+--------+
| LOCATIONNAME | AMOUNT |
+--------------+--------+
| CHENNAI HUB  | 350.00 |
+--------------+--------+
1 row in set (0.00 sec)
```

54. Find all couriers whose weight is greater than the weight of all couriers sent by a specific sender (e.g., 'SenderName'):

```
mysql> SELECT * FROM COURIER WHERE WEIGHT > ALL(SELECT WEIGHT FROM COURIER WHERE SENDERNAME = 'RITHIKA');
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+
| COURIERID | SENDERNAME | SENDERADDRESS | RECEIVERNAME | RECEIVERADDRESS | WEIGHT | STATUS     | TRACKINGNUMBER | DELIVERYDATE |
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+
|       101 | ELAKKIYA   | CHENNAI       | ROJA         | DELHI           |   3.25 | IN TRANSIT | TRK100001      | 2025-06-18   |
|       104 | SHOBITHA   | BANGALORE     | TANUJA       | KOLKATA         |   4.75 | PENDING    | TRK100004      | 2025-06-25   |
+-----------+------------+---------------+--------------+-----------------+--------+------------+----------------+--------------+
2 rows in set (0.00 sec)
```